AMENDMENTS TO THE CLAIMS

1. (Previously Presented) A computer-implemented method comprising:

providing a graphical user interface for defining a function to be used in a graphical representation of a finite state machine, where the graphical representation is an executable model of the finite state machine and includes at least one state and at least one transition;

representing the function graphically such that the function is graphically represented separately from the at least one state and at least one transition in the graphical representation of the finite state machine, wherein the function that is represented graphically has a function prototype that specifies a syntax for invoking the function, the function prototype specifying a function name for the function, and the function is defined in a graphical language; and

calling the function that is represented graphically by the function name according to the syntax specified by the function prototype from within the graphical representation of the finite state machine.

2. (Previously Presented) The method of claim 1 wherein the graphical function further includes a function block.

3. (Canceled)

4. (Previously Presented) The method of claim 1 wherein the graphical function further includes a function flow diagram that graphically defines a procedure performed by the function.

5. (Previously Presented) A computer-implemented method, said method comprising:

providing a graphical user interface for defining a function to be used in a graphical representation of a finite state machine, where the graphical representation is an executable model of the finite state machine and includes at least one state and at least one transition;

representing the function graphically such that the function is graphically represented separately from the at least one state and the at least one transition in the graphical representation of the finite state machine,

wherein the function is represented graphically as a diagram comprising graphical elements and has a function prototype that specifies a syntax for invoking the function, the function prototype specifying a function name for the function; and

calling the function that is represented graphically by the function name according to the syntax specified by the function prototype from within the graphical representation of the finite state machine.

6. (Previously Presented) The method of claim 1 further comprising modifying the function through graphical diagramming.

7. – 11. (Canceled)

12. (Currently Amended) A computer program product, stored in a computer readable storage medium, comprising instructions to cause a computer to:

receive input defining a graphical function for use in a finite state machine the graphical function having a function prototype that specifies a syntax for invoking the graphical function, the function prototype specifying a function name for the graphical function; and

use the graphical function in a simulation of a system represented by the finite state machine, wherein the instructions to use the graphical function further comprise instructions to call the graphical function by the function name according to the syntax specified by the function prototype from at least one state or transition in the finite state machine, the graphical function being represented graphically such that the graphical function is graphically represented separately from the at least one state or ~~and the at least one~~ transition in the finite state machine.

13. (Previously Presented) The computer program product of claim 12 wherein the input defining the graphical function is entered into a function block.

14. (Canceled)

15. (Previously Presented) The computer program product of claim 12 wherein the input comprises a function flow diagram that graphically defines a procedure performed by the function.

16. (Previously Presented) The computer program product of claim 15 wherein the function flow diagram is comprised of graphical elements.

17. (Previously Presented) A system for modeling at least one finite state machine, said system comprising:

a computer comprising a graphical user interface, a memory, a storage, and at least one input device;

means to receive input to define a graphical function, the graphical function having a function prototype that specifies a syntax for invoking the graphical function, the function prototype specifying a function name for the graphical function;

means to represent the graphical function as an executable state flow diagram; and

means to call the graphical function by the function name according to the syntax specified by the function prototype from the at least one finite state machine in a simulation of the at least one finite state machine, the at least one finite state machine including at least one state and at least one transition, the graphical function being represented graphically such that the graphical function is graphically represented separately from the at least one state and the at least one transition in the finite state machine.

18. (Previously Presented) The system of claim 17 wherein the input to define the graphical function is entered into a function block.

19. (Canceled)

20. (Canceled)

21. (Previously Presented) The system of claim 17 wherein the input comprises a function flow diagram that graphically defines a procedure performed by the graphical function.

22. (Previously Presented) The system of claim 21 wherein the function flow diagram is comprised of graphical elements.

23. (Previously Presented) The system of claim 21 further comprising means for hiding the display of the function flow diagram based upon input.

24. (Previously Presented) A method of operating a data processing system having a graphical user interface, said method comprising:

creating a graphical representation of a finite state machine and a graphical representation of a function for use in the graphical representation of the finite state machine, the function having a function prototype that specifies a syntax for invoking the function, the function prototype specifying a function name for the function, the graphical representation of the finite state machine including at least one state and at least one transition, the graphical representation of the function being represented graphically such that the function is graphically represented separately from the at least one state and the at least one transition in the graphical representation of the finite state machine;

simulating a system represented by the finite state machine, wherein the graphical representation of the finite state machine is an executable model of the system; and

calling the function by the function name according to the syntax specified by the function prototype from the executable model of the system during the act of simulating the system represented by the finite state machine.

25. (Canceled)

26. (Canceled)

27. (Canceled)

28. (Previously Presented) The method of claim 24 further comprising shadowing the function, wherein shadowing comprises using in a function invocation a function definition closest to a point of invocation of the function in a state diagram hierarchy.

29. (Previously Presented) The method of claim 24 wherein the function is exportable by a state chart and may be invoked anywhere in the finite state machine in which the chart appears, including other charts that define the finite state machine.

30. (Previously Presented) The method of claim 24 wherein simulating the system represented by the finite state machine further comprises computer code generation.

31. (Canceled)

32. (Previously Presented) A computer readable storage medium having encoded thereon:

instructions for causing a computer system to receive through a graphical user interface a graphical representation of a finite state machine and a graphical representation of a function for use in the graphical representation of the finite state machine, the function having a function prototype that specifies a syntax for invoking the function, the function prototype specifying a function name for the function, the graphical representation of the finite state machine including at least one state and at least one transition, the function being represented graphically such that the function is graphically represented separately from the at least one state and the at least one transition in the graphical representation of the finite state machine; and

instructions for simulating a system represented by the finite state machine, where the graphical representation is an executable model of the system; and

instructions for calling the function by the function name according to the syntax specified in the function prototype from at least one place in the executable model during the system simulation.

33. (Canceled)

34. (Previously Presented) In an electronic device, a method of graphically representing an event-driven system, said method comprising:

providing one or more block components representing one or more states in an executable model;

providing one or more transition components representing transitions between the one or more states; and

providing a function having a function prototype that specifies a syntax for invoking the function, the function prototype specifying a function name for the function, said function comprising at least two graphical components and being referenced by at least one of the states

or at least one of the transitions to call the function by the function name according to the syntax specified by the function prototype at the at least one of the states or the at least one of the transitions, the function being represented graphically such that the function is graphically represented separately from the at least one of the states and the at least one of the transitions in the executable model.

35. (Previously Presented) The method of claim 34, wherein the function accepts at least one argument and returns at least one result.

36. (Previously Presented) The method of claim 34, further comprising invoking the function at a second one of the one or more transition components or one or more block components.

37. (Previously Presented) The method of claim 34, further comprising specifying data properties of the function.

38. (Previously Presented) The method of claim 34, further comprising associating a data item with the function.

39. (Previously Presented) The method of claim 34, wherein the function comprises a graphical function.

40. (Previously Presented) The method of claim 34, wherein the function has a plurality of configurable properties.

41. (Canceled)

42. (Previously Presented) The method of claim 34, further comprising providing a shadowing function, wherein shadowing comprises using in a function invocation a function definition proximally closest to a point of invocation of the function in a state diagram hierarchy.

43. (Previously Presented) In a graphical representation environment, a system for graphically representing an event-driven system, said system comprising:

one or more block components representing one or more states in an executable model;

one or more transition components representing transitions between the one or more block components representing the one or more states; and

a component representing a graphical function having a function prototype that specifies a syntax for invoking the function, the function prototype specifying a function name for the graphical function, and referenced by at least one of the states or at least one of the transitions to call the graphical function by the function name according to the syntax specified by the function prototype at one of the states or one of the transitions, the graphical function being represented graphically such that the graphical function is graphically represented separately from the at least one of the states and the at least one of the transitions in the executable model.

44. (Previously Presented) The system of claim 43, wherein the graphical function accepts at least one argument and returns at least one result.

45. (Previously Presented) The system of claim 43, wherein at least a subset of the one or more block components representing the states and the one or more transition components can invoke the graphical function.

46. (Previously Presented) The system of claim 43, further comprising means for specifying data properties of the graphical function.

47. (Previously Presented) The system of claim 43, further comprising means for associating a data item with the graphical function.

48. (Previously Presented) The system of claim 43, wherein the component representing the graphical function is referenced by one more of: at least one of the states or at least one of the transitions.

49. (Previously Presented) The system of claim 43, wherein the graphical function has a plurality of configurable properties.

50. (Canceled)

51. (Previously Presented) The system of claim 43, further comprising means for providing a shadowing function, wherein shadowing comprises using in a function invocation a function definition proximally closest to a point of invocation of the function in a state diagram hierarchy.

52. (Previously Presented) A computer-readable storage medium for use in a graphical representation environment on an electronic device, the medium holding instructions executable using the electronic device for graphically representing an event-driven system, said instructions comprising instructions for:

providing one or more block components representing one or more states in an executable model;

providing one or more transition components representing transitions between the one or more block components representing the one or more states; and

providing a block component representing a graphical function having a function prototype that specifies a syntax for invoking the graphical function, the function prototype specifying a function name for the graphical function and referenced by at least one of the states or at least one of the transitions to call the graphical function by the function name according to the syntax specified by the function prototype at one of the states or one of the transitions during execution of the event-driven system, the graphical function being represented graphically such that the graphical function is graphically represented separately from the at least one of the states and the at least one of the transitions in the executable model.

53. (Previously Presented) The computer-readable storage medium of claim 52, wherein the graphical function accepts at least one argument and returns at least one result.

54. (Previously Presented) The computer-readable storage medium of claim 52, wherein the one or more transition components can invoke the graphical function.

55. (Previously Presented) The computer-readable storage medium of claim 52, further comprising instructions for accepting ~~user~~ input specifying data properties of the graphical function.

56. (Previously Presented) The computer-readable storage medium of claim 52, further comprising instructions for associating a data item with the graphical function.

57. (Previously Presented) The computer-readable storage medium of claim 52, wherein the graphical function comprises two or more graphical elements.

58. (Previously Presented) The computer-readable storage medium of claim 52, wherein the graphical function has a plurality of configurable properties.

59. (Canceled)

60. (Previously Presented) The computer-readable storage medium of claim 52 further comprising instructions for providing a shadowing function, wherein shadowing comprises using in a function invocation a function definition proximally closest to a point of invocation of the graphical function in a state diagram hierarchy.

61. (Previously Presented) A computer-implemented method for modeling a system using a graphical block diagram environment, said method comprising:

graphically representing a function having a function prototype that specifies a syntax for invoking the function, the function prototype specifying a function name for the function for use in an executable model within the graphical block diagram environment, the executable model including at least one state and at least one transition, the function being represented graphically such that the function is graphically represented separately from the at least one state and the at least one transition in the executable model; and

textually referencing the graphically represented function by the function name according to the syntax specified by the function prototype within the model to cause an invocation of the graphically represented function during execution of the model.

62. (Previously Presented) The computer-implemented method of claim 61, wherein the model is represented as a finite state machine.

63. (Previously Presented) The computer-implemented method of claim 62, wherein the finite state machine is a hierarchical finite state machine.

64. (Previously Presented) The computer-implemented method of claim 62 further comprising:

associating the graphically represented function with at least one state or transition within the finite state machine.

65. (Previously Presented) The computer-implemented method of claim 61, wherein the graphically represented function is represented as at least one of a finite state machine, a state flow diagram, a function flow diagram, and a graphical block diagram model.

66. (Previously Presented) A computer-readable storage medium holding instructions executable using an electronic device for modeling a system using a graphical block diagram environment, said instructions comprising instructions for:

graphically defining a function having a function prototype that specifies a syntax for invoking the function, the function prototype specifying a function name for the function for use in an executable model within the graphical block diagram environment, the executable model including at least one state and at least one transition, the function being represented graphically such that the function is graphically represented separately from the at least one state and the at least one transition in an executable model; and

textually referencing the graphically represented function by the function name according to the syntax specified by the function prototype within the model to cause an invocation of the graphically represented function during execution of the model.

67. (Previously Presented) The computer-readable storage medium of claim 66, wherein the model is represented as a finite state machine.

68. (Previously Presented) The computer-readable storage medium of claim 67 further comprising instructions for:
associating the graphically represented function with at least one state or transition within the finite state machine.

69. (Previously Presented) The computer-readable storage medium of claim 66, wherein the graphically represented function is represented as at least one or a combination of: a finite state machine, a state flow diagram, a function flow diagram, and a graphical block diagram model.

70. (Previously Presented) A computer-implemented system for modeling using a graphical block diagram environment, said system comprising:

means for representing a function having a function prototype that specifies a syntax for invoking the function, the function prototype specifying a function name for the function, the function being defined graphically for use in an executable model within the graphical block diagram environment, the executable model including at least one state and at least one transition, the function being represented graphically such that the function is graphically represented separately from the at least one state and the at least one transition in the executable model; and

means for textually referencing the function defined graphically by the function name according to the syntax specified by the function prototype within the model to cause an invocation of the function during execution of the model.

71. (Previously Presented) The system of claim 70, wherein the executable model is represented as a finite state machine.

72. (Previously Presented) The system of claim 71 further comprising:

means for associating the graphically represented function with at least one state or transition within the finite state machine.

73. (Previously Presented) The system of claim 70, wherein the graphically represented function is represented as at least one or a combination of a finite state machine, a state flow diagram, a function flow diagram, and a graphical block diagram model.

74. (Previously Presented) A graphical block diagram modeling system comprising:

a graphical function for use in an executable model, the graphical function having a function prototype that specifies a syntax for invoking the graphical function, the function prototype specifying a function name for the graphical function, wherein at least a subset of commands of the graphical function are defined through a graphical representation, the executable model including at least one state and at least one transition, the graphical function being represented graphically such that the graphical function is graphically represented separately from the at least one state and the at least one transition in the executable model; and

a graphical representation of the model including a textual reference of the graphical function by the function name according to the syntax specified by the function prototype within the graphical representation of the model to cause an invocation of the graphical function during an execution of the model.

75. (Previously Presented) The system of claim 74, wherein the model is represented as a finite state machine.

76. (Previously Presented) The system of claim 75, wherein the finite state machine is a hierarchical finite state machine.

77. (Previously Presented) The system of claim 75, wherein the finite state machine further comprises:

at least one state or transition associated with the graphical function.

78. (Previously Presented) The system of claim 74, wherein the graphical function is represented as at least one or a combination of: a finite state machine, a state flow diagram, a function flow diagram, and a graphical block diagram model.